

5

AN INTERACTIVE PATIENT COMMUNICATION DEVELOPMENT
SYSTEM FOR REPORTING ON PATIENT HEALTHCARE MANAGEMENT

10

FIELD OF THE INVENTION

The present invention relates generally to a modular interactive development system and method for reporting on patient management, and in particular to an automated content delivery program able to connect remote users across independent platforms to a central database of libraries whereby a patient's health can be scored dynamically.

20 BACKGROUND OF THE INVENTION

This invention relates to the field of health management, particularly to an automated interactive system and method for reducing the risk associated with a monitored client.

For example, the know art includes a number of health-management systems for providing outpatient services to patients with chronic health conditions such as asthma and diabetes. However, these systems are incapable of administering a treatment protocol responsive to the patient's current profile and of updating the profile in response to the administered protocol.

30

SUMMARY OF THE INVENTION

This invention presents a flexible and scalable system in content development for patient management healthcare. Due to the modular object oriented-structure, individual content modules ("dialogs") can be mixed into an
5 unlimited number of updateable customized programs, addressing individual as well as co-existing disease states ("co-morbid") in any combinations, and with automated content variation for improved patient compliance. A dialog is the smallest content object in the FlexCube content structure. Its content addresses issues related to a unique set of symptoms, behaviors or knowledge related to a
10 specific aspect of managing a certain disease referred to as an aspect of care.

In its basic format, each dialog contains questions related to signs and symptoms, behaviors and knowledge with answers categorized as high, medium or low risk answers. For each answer there is a relevant follow up, which can be a teaching statement, an acknowledgment, a motivational statement or a new
15 question that will explore the patient's condition in more depth. While the logical branching within a dialog is driven by patient answers, no dependency exists between individual dialogs.

Dialogs are located in a common pool organized by library. From this library each individual dialog is referenced for participation (appearance) in
20 programs and daily sessions. A dialog's behavior in a program (schedule, position, reporting) is defined at the time of the dialog creation or it is custom defined during the program content selection process. In this way dialogs maintain their integrity while being used and re-used in several client programs. They combine freely with other dialogs in user defined program selections, allowing an unlimited
25 combination of aspects of care and co-existing diseases. Finally, they are easily accessible for revisions and updates.

The present invention provides an object-oriented dialog and modular toolkit structure that enhances quality control options. Also included are the centrally located content objects that offer overview and tracking of the currently

- active content, global error correction and global update of content to current standards of care. Because the present invention splits up interfaces for content creation and content selection into separate modules, the present invention exercises control over customer's access to content development in compliance
- 5 with current and future Federal Drug Administration labeling. Finally the system's structure limits logical branching errors to within a dialog, thereby offering a more robust and less error prone system overall.

Since the content of a dialog and the output of a dialog is related and mapped to a specific aspect of care, the user will have the power and flexibility to

10 model risk evaluation and outcomes reporting around custom selected aspects of care.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing aspects and many of the attendant advantages of this

15 invention will become more readily appreciated as the same becomes better understood by reference to the following detailed description, when taken in conjunction with the accompanying drawings, wherein:

FIGURE 1 is a block diagram depicting a system's compositional and referenced components;

20 FIGURE 2 is a flow chart diagram depicting the overview of dialog creation;

FIGURE 3 is a block diagram depicting an interdependent characteristics (operators) of a dialog;

25 FIGURE 4 is flow chart depicting the steps in creating and storing of content data from a dialog;

FIGURE 5 is a flow chart diagram depicting the creation of the programming statements using a Dialog Editor Platform;

FIGURE 6 is a block diagram illustrating the three dimensional aspects of the dynamically determined risk state output scale;

FIGURE 7 is a flowchart depicting the creation of programs using a Program Composer User Interface;

FIGURE 8 is a flow chart depicting a Linker User Interface; and

FIGURE 9 is a flow chart depicting a Reporter User Interface.

5

DETAILED DESCRIPTION OF PREFERRED EMBODIMENT

The present invention includes an object-oriented content structure in which the smallest content object, a care specific dialog, is located in a central library from where its characteristics (operators) are composed and referenced by a 10 modular set of tools located at a client computer.

FIGURE 1 is a block diagram depicting a system 10's compositional and referenced components. Compositionally, the system 10 relies on four system components for dialog or program creation. Additionally, FIGURE 1 illustrates two other system components that interact with the referenced components of the 15 system. A dialog Composer 20, further referenced in FIGURE 2, which is used to author dialog content by an aspect of care. A Program Composer 30, further referenced in FIGURE 7, is a user interfaced click and drag assembly platform for composing programs (a virtual content defined collection of dialogs). On a computer desktop, content dialogs are selected (referenced) for use in 20 disease/client specific programs, with program specific tagging of individual dialog attributes related to frequency (scheduling) and reporting. A Program Patient Linker 40 is a user interface integrated into the desktop on which patients are assigned to programs. During the assignment process patient identification and patient specific metrics are added to the program. A Care Reporter 50, further 25 referenced in FIGURE 9, is a user interface for easy patient result lookup, triage and trend reports. Reporting requirements set in the Program Composer 30 determine which reports are displayed.

Compositional elements of the system 10 reference either one or both of the two remaining components of the system depicted in FIGURE 1. A Program

Scheduler 60, further referenced in FIGURE 8, is an engine for automated scheduling of dialogs based on attributes set in the Program Composer 30, and A Dialog Library 70. The Dialog Library is the principal central location of dialog content units. Dialogs are organized into body system labeled sub-libraries and

5 stored within the Dialog Library 70.

The structure of the system is developed from the integration of the four compositional components as referenced above with the two referenced components and begins with the creation of dialogs in the Dialog Composer 20 as depicted FIGURE 2.

10 FIGURE 2 is a flow chart diagram depicting the overview of dialog creation and is referenced with more particularity in FIGURE 4. Referring to FIGURE 2, a patient 100 reports on a specific aspect of care 110 (i.e., foot care in a Diabetes Structure) that is addressed by a dialog 125, the smallest content structure of the system, from a disease specific library 120. The basic format of

15 each dialog includes questions 130 related to patient self-management behaviors 132, patient-reportable symptoms 134, or patient knowledge 136. Each question provides a choice for an answer (“output variable”) 140 that falls into one of three risk categories; high 142 medium 144 and low risk 146. For each risk category there is an associated follow up 150 which is a teaching statement 152, a

20 motivational statement 154 or a new question 156 that explores the patient's condition in more depth.

While the logical branching within a dialog depends on output variables, no dependency exists between individual dialogs. Dependencies for dialogs exist outside the dialog structure in related operators.

25 FIGURE 3 is a block diagram depicting the interdependent characteristics (operators) of a dialog 300 in the system matrix. The interdependent characteristics include a Name Label 310 for the aspect of care addressed, a Library 320 that houses a body system specific Localization 325, client specific Programs 330 in which the dialog is being used (referenced), a Schedule frequency

340 by which the dialog is being displayed to a patient in a specific program, definition of Reporting requirements 350, and Patient Identification information 360 and metrics of each individual appliance to which the dialog is assigned.

- The user interface is easy to use due to the simplicity of program structure
- 5 in which the user is able to interface with the program and dialog composition aspects of the system. Simply using drag and drop content selection procedures based on a medical decision creates a process familiar to the user. The user decides what aspects of care are relevant for a given program or for an individual patient and in most cases simply selects existing content based on that decision. In
- 10 all steps of dialog composition, certain steps are taken to make available the dialog in a content library.

- FIGURE 4 is a flow chart depicting the steps in creating and storing of content data from a dialog, a user's first task is to name the dialog-to-be-created as depicted in block 400. Next, the user defines the library section of block 410, in
- 15 which the dialog will reside. The user then identifies an aspect of care at block 420 to which the dialog will primarily refer. Once the naming conventions are assigned and the aspect of care is chosen, the user creates dialog programming statements at block 430, in a graphical programming environment as embodied in FIGURE 5. New dialog content is then stored in an appropriate user library at block 440.

- 20 The user who has access to create new content does so using a simple dialog composer as embodied in FIGURE 5. FIGURE 5 is a diagram depicting the creation components of a dialog Editor Platform. First, a user is presented with a palette 500 of programming statements that are represented as graphic symbols (icons) that can be dragged from the palette of available statements into a dialog
- 25 construction platform 505. In a typical embodiment of the present invention, the user drags a start question icon 510 and a three pronged answer icon 520 from an icon palette down to the construction platform 500. The user then activates a dialog box for each icon by clicking on it with a mouse and specifying a question associated with that particular icon, for example, a Start Question Dialog 515.

- Next, in an Answer Dialog 524, the user enters three answer options relative to the start question and assigns a raw risk value to each answer 526. The risk values are assigned from high to low with a corresponding text answer. “Yes” equals low risk and “no” equals high risk and “medium” equals somewhere in the middle of
- 5 low and high risk. Follow up questions icons 530 are dragged onto the construction platform along with an associated answer icon 540. An answer dialog 545 is then prepared. Clicking on the output icon 550, the user activates the output dialog box 555. Here the user defines risk state output 558 in detail, further depicted with more particularity in FIGURE 5, defining the position of the answer
- 10 relative to the axis of the risk cube. At any time during or after the dialog creation process, the user can review the dialog created, using a simulation interface to an appropriate appliance or in the alternative, the user can review the actual dialog content in a text only overview window. Once all the follow up questions, answers and output dialogs are formulated and put onto the construction platform 525, the
- 15 newly created dialogs are stored in a user library 560 from where it can be referenced for participation in any user defined care management program or for later updating or editing.

FIGURE 6 is a block diagram illustrating the three dimensional aspects of the dynamically determined risk state output scale which in the Dialog Composer,

20 FIGURE 5, is referenced at block 558. The X-axis 610 scales whether the answer to a question dialog sets the risk at a certain risk level on a 9 point risk scale or whether the answer moves the patient risk state in a certain direction and by how much, thereby creating an accumulated risk profile. Additionally, the answer to a dialog is incorporated as a value in a mathematically calculated risk state that may incorporate other answers as well, creating a composite, weighted risk state. The Y-axis 620 refers to the actual aspect of care in which the risk will be incorporated. The Z-axis 630 incorporates the expression of risk 530, i.e., whether the risk is assigned to a sign or symptom 632, a behavior 634, or a knowledge expression 636. This dynamic model allows for very sophisticated risk profiling

including risk trend alerts, composite risk profiling by aspects of care and profiling by risk expression. The dynamic risk "foot prints" available at any time can serve as triggers for automated content selection.

- Once dialogs are named, created and assigned to an aspect of care and the
- 5 risk output is assigned to the appropriate dialog, a user of the system can then use the Program Composer 30 to create the program that eventually is assigned to a patient.

FIGURE 7 is a flowchart depicting the creation of "programs" using the Program Composer User Interface ("UI"). The UI is a platform for selecting library resident Dialogs created as depicted in FIGURE 6, for participation in user-defined care management programs. In a typical embodiment of the present invention, the first step is to name the future program block 700. Next, at block 710, a user selects the disease libraries from which the program dialogs are created. Simultaneously, at block 720, the user checks the Utilities Library to add dialogs to the program that are not disease specific like generic greetings. This gives the user access to the detailed content of both of these libraries organized by aspects of care and their respective dialogs. Creating the program is now a simple task of adding dialogs to the program list, see block 730, and at block 740 to define the delivery of the dialogs as a user can choose specific delivery of the

10 dialogs on a daily 750, weekly 752, or any other 754 programmed timed basis. Additionally, at block 742, a user checks the priority of dialogs to set parameters necessary for the correct scheduling of the dialogs in the program. Options are to force the scheduler to include the dialog block 744, or to assign dialogs as fillers, block 746. The later could be the case, for example, with trivia type dialogs,

15 entertainment dialogs etc. Also, the user has the opportunity to decide the placement of dialogs in daily sessions. Greetings, for example, should be checked as "always first." The user can review the complete created program using the "View Selection" link, block 760. Using a very simple interface, the user has now created a totally custom made program. At block 770, the program is now

20

available for assignment to any of the user's patients or for later modification by the user by adding or deleting dialogs. The present invention embodies the assignment by way of a Linker User Interface ("Linker UI") as depicted in FIGURE 8.

- 5 FIGURE 8 is a flow chart depicting the Linker UI, which is a platform for assigning or "linking" care management programs to patient populations or to individual patients. The first step at block 800 is to retrieve patient's name(s) to be used on the work platform through a filtering or sorting procedure defined by the user. Next, at block 810, the user marks the patient(s) and the care management
- 10 10 program to be assigned. Finally the user creates the "Link" to activate a dialog box that allows the user to specify a time frame in which the program will run for the selected patient(s), block 820. Should the user wish to link the patient to other programs all that is needed is to repeat the process. To process the linking of an entire population or part of a population a user selects all patients, block 800, and
- 15 15 assigns all of them, block 810, to a program.

- The last step in the creation of a system program is the creation of a Reporter User Interface ("Reporter UI") which creates patient reports specific to patient results that in turn can initiate program actions based on those results. FIGURE 9 is a flow chart depicting the Reporter UI and the creation of reports.
- 20 20 The layout of the Reporter UI is completely consistent with that of the Linker UI depicted in FIGURE 8. First a user retrieves patient names through a filtering process, block 902. The user filters, at block 900, names through the programs by either risk search, block 904, the aspects of care, block 905, within each program, or the risk expression, block 906, as defined as a symptom, behavior or knowledge, block 908, factor. This is done to allow a user to trend a risk profile, block 910, for the patient in the aspect of care where the patient has scored, for example, a high-risk profile as depicted in FIGURE 6. A user can configure the Reporter UI to display block 920 the actual answers or results that led to the exemplified high-risk profile. Lastly, at block 930, a patient is assigned to a program

HERO-1-1111AP

based on the risk profile or Aspect of Care. Reports assigned to patients can now for example, allow the user to see details for each aspect of care, order a report printed or write a note that will be associated with a linked event.

While this invention has been described in terms of several preferred 5 embodiments, there are alterations, permutations, and equivalents that fall within the scope of this invention. It is therefore intended that the following appended claims be interpreted as including all such alterations, permutations, and equivalents as fall within the true spirit and scope of the present invention.

10

10110011001011001